(19)日本国特許庁（ＪＰ）　(12)**公開特許公報（Ａ）**　(11)特許出願公開番号
特開2003－169216
（P2003－169216A）
(43)公開日　平成15年６月13日(2003.6.13)

(51)Int.Cl.[7]	識別記号	ＦＩ		テーマコード*（参考）	
Ｈ０４Ｎ　1/41		Ｈ０４Ｎ	1/41	Ｂ	５Ｃ０５９
Ｈ０３Ｍ　7/30		Ｈ０３Ｍ	7/30	Ａ	５Ｃ０７８
Ｈ０４Ｎ　7/30		Ｈ０４Ｎ	7/133	Ｚ	５Ｊ０６４

審査請求　未請求　請求項の数26　ＯＬ　（全 24 頁）

(21)出願番号　特願2001－364894(P2001－364894)

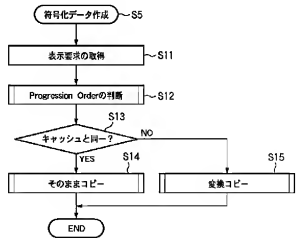
(22)出願日　平成13年11月29日(2001.11.29)

(71)出願人　000001007
キヤノン株式会社
東京都大田区下丸子３丁目30番２号
(72)発明者　榎田　幸
東京都大田区下丸子３丁目30番２号　キヤノン株式会社内
(72)発明者　石川　智恵
東京都大田区下丸子３丁目30番２号　キヤノン株式会社内
(74)代理人　100076428
弁理士　大塚　康徳　（外３名）

最終頁に続く

(54)【発明の名称】　符号化データの作成方法及びその装置

(57)【要約】
【課題】　断片化された符号化データを受信してキャッシュしておき、そのキャッシュした符号化データを用いて表示要求に応じた符号化データを作成する。
【解決手段】　断片化された符号化データを受信してメモリに格納しておき、ユーザの表示要求に基づいてProgression Orderを判断し（Ｓ１２）、その判断に基いてヘッダ情報を変更し、その変更したヘッダ情報に基づいて、メモリに該当する符号化データが格納されているかどうかを判定し、メモリに格納されていると判定された断片化された符号化データを読み出して、その表示要求に適したProgression Orderに変換するとともに、メモリに格納されていない符号化データにはＺＬＰを適用してＪＰＥＧ２０００準拠の符号化データを作成する。



【特許請求の範囲】

【請求項１】　断片化された符号化データを受信してメモリに格納する格納工程と、
ユーザの表示要求を取得する取得工程と、
前記取得工程で取得した前記表示要求に基づいてProgression Orderを判断する判断工程と、
前記判断工程での判断に基いてヘッダ情報を変更する変更工程と、
前記変更工程で変更された前記ヘッダ情報に基づいて、前記メモリに該当する符号化データが格納されているかどうかを判定する判定工程と、
前記判定工程において前記メモリに格納されていると判定された前記断片化された符号化データを読み出して前記表示要求に適したProgression Orderに変換するとともに、前記メモリに格納されていない符号化データにはダミー符号を適用してＪＰＥＧ２０００準拠の符号化データを作成する工程と、を有することを特徴とする符号化データの作成方法。

【請求項２】　前記断片化された符号化データは、ネットワークを介して送信されるＪＰＥＧ２０００符号化データであり、前記符号化データはパケット単位に送信されることを特徴とする請求項１に記載の符号化データの作成方法。

【請求項３】　前記ダミー符号は、ＪＰＥＧ２０００で規定されているzerolength packetのデータであることを特徴とする請求項１に記載の符号化データの作成方法。

【請求項４】　前記変更工程では、画像全体のヘッダ情報を変更することを特徴とする請求項１に記載の符号化データの作成方法。

【請求項５】　前記変更工程では、タイル単位のヘッダ情報を変更することを特徴とする請求項１に記載の符号化データの作成方法。

【請求項６】　前記判断工程では、前記符号化データの解像度レベル及びレイヤ数と、前記表示要求の解像度レベル及びレイヤ数とに基づいて判断することを特徴とする請求項１に記載の符号化データの作成方法。

【請求項７】　断片化された符号データを受信してメモリに格納する格納工程と、
ユーザの表示要求を取得する取得工程と、
前記取得工程で取得した前記表示要求から表示に最低限必要とされる符号データを特定する特定工程と、
前記特定工程で特定された符号データを前記メモリから読み出し、その他の符号データをダミー符号に置き換えることにより、前記最低限必要とされる符号データを有する符号化データを作成する作成工程と、ことを特徴とする符号化データの作成方法。

【請求項８】　前記断片化された符号化データは、ネットワークを介して送信されるＪＰＥＧ２０００符号化データでパケット単位に送信されることを特徴とする請求項７に記載の符号化データの作成方法。

【請求項９】　前記ダミー符号は、ＪＰＥＧ２０００で規定されているzerolength packetのデータであることを特徴とする請求項８に記載の符号化データの作成方法。

【請求項１０】　前記最低限必要とされる符号化データの単位は、タイル単位であることを特徴とする請求項７に記載の符号化データの作成方法。

【請求項１１】　前記最低限必要とされる符号化データの単位は、パケット単位であることを特徴とする請求項７に記載の符号化データの作成方法。

【請求項１２】　前記特定工程では、前記メモリに格納されているか否かに拘わらず、前記表示要求に基づいて特定することを特徴とする請求項７に記載の符号化データの作成方法。

【請求項１３】　断片化された符号化データを受信して格納する格納手段と、
ユーザの表示要求を取得する取得手段と、
前記取得手段で取得した前記表示要求に基づいてProgression Orderを判断する判断手段と、
前記判断手段での判断に基いてヘッダ情報を変更する変更手段と、
前記変更手段で変更された前記ヘッダ情報に基づいて、前記格納手段に該当する符号化データが格納されているかどうかを判定する判定手段と、
前記判定手段において前記格納手段に格納されていると判定された前記断片化された符号化データを読み出して前記表示要求に適したProgression Orderに変換するとともに、前記格納手段に格納されていない符号化データにはダミー符号を適用してＪＰＥＧ２０００準拠の符号化データを作成する手段と、を有することを特徴とする符号化データの作成装置。

【請求項１４】　前記断片化された符号化データは、ネットワークを介して送信されるＪＰＥＧ２０００符号化データであり、前記符号化データはパケット単位に送信されることを特徴とする請求項１３に記載の符号化データの作成装置。

【請求項１５】　前記ダミー符号は、ＪＰＥＧ２０００で規定されているzero length packetのデータであることを特徴とする請求項１３に記載の符号化データの作成装置。

【請求項１６】　前記変更手段は、画像全体のヘッダ情報を変更することを特徴とする請求項１３に記載の符号化データの作成装置。

【請求項１７】　前記変更手段は、タイル単位のヘッダ情報を変更することを特徴とする請求項１３に記載の符号化データの作成装置。

【請求項１８】　前記判断手段は、前記符号化データの解像度レベル及びレイヤ数と、前記表示要求の解像度レベル及びレイヤ数とに基づいて判断することを特徴とす

る請求項１３に記載の符号化データの作成装置。

【請求項１９】　断片化された符号データを受信して格納する格納手段と、ユーザの表示要求を取得する取得手段と、

前記取得手段で取得した前記表示要求から表示に最低限必要とされる符号データを特定する特定手段と、

前記特定手段で特定された符号データを前記格納手段から読み出し、その他の符号データをダミー符号に置き換えることにより、前記最低限必要とされる符号データを有する符号化データを作成する作成手段と、ことを特徴とするデータの作成装置。

【請求項２０】　前記断片化された符号化データは、ネットワークを介して送信されるＪＰＥＧ２０００符号化データでパケット単位に送信されることを特徴とする請求項１９に記載の符号化データの作成装置。

【請求項２１】　前記ダミー符号は、ＪＰＥＧ２０００で規定されているzero length packetのデータであることを特徴とする請求項２０に記載の符号化データの作成装置。

【請求項２２】　前記最低限必要とされる符号化データの単位は、タイル単位であることを特徴とする請求項１９に記載の符号化データの作成装置。

【請求項２３】　前記最低限必要とされる符号化データの単位は、パケット単位であることを特徴とする請求項１９に記載の符号化データの作成装置。

【請求項２４】　前記特定手段では、前記格納手段に格納されているか否かに拘わらず前記表示要求に基づいて特定することを特徴とする請求項１９に記載の符号化データの作成装置。

【請求項２５】　請求項１乃至１２のいずれか１項に記載の符号化データの作成方法を実行するプログラムを記憶したことを特徴とするコンピュータにより読取り可能な記憶媒体。

【請求項２６】　請求項１乃至１２のいずれか１項に記載の符号化データの作成方法を実行することを特徴とするプログラム。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、例えばネットワークを介して受信した断片的な画像データを、一つの画像ファイルに作成し直して符号化する符号化データの作成方法及びその装置に関するものである。

【０００２】

【従来技術】インターネット上では、WWWサーバにWeb（ウエブ）ブラウザからアクセスして文書データや画像データ等を閲覧することが盛んに行われている。この仕組みはインターネット上に情報を公開するWWWサーバと、その情報を閲覧するクライアントを含み、各クライアントはウエブブラウザを使用して、そのサーバにより公開された情報を閲覧することができる。このWWWサーバには、ホームページといわれる公開したい情報をHTMLで記述した文書があり、それをクライアント側のウエブブラウザがアクセスしてクライアントのコンピュータに表示する。またクライアント側のウエブブラウザは、表示しているページ内のリンクを辿っていくことにより、自分が必要な情報を得ることができる。更に、サーバが管理しているファイルをダウンロードする方法として、File Transfer Protocol（以下、ＦＴＰと略す）という方法がある。このＦＴＰとは、ネットワークを通して、サーバ上にあるファイルを一度にクライアント・コンピュータに転送する仕組みである。

【０００３】また画像ファイルへ断片的にアクセスして表示するためのプロトコルとしてFlashpix／ＩＩＰがある。このＩＩＰは、Flashpixという、画像データファイルのフォーマットに最適なプロトコルになっており、画像データの部分アクセスの単位は、このFlashpixのタイル単位に行うものである。

【０００４】一方、このＩＩＰをそのままＪＰＥＧ２０００に適用した場合を考えると、ＪＰＥＧ２０００では、各スケイラビリティの符号化データは、そのスケイラビリティより１つ下のスケイラビリティのデータからの差分データで構成されている。そのためクライアント側で受信した断片的な符号化データをキャッシュしておき、全符号化データをデコーダに引き渡して最初から再デコードする方法と、デコーダを途中で止めておいて今回受信した符号化データをデコーダに渡し、前回の続きからデコードする方法がある。

【０００５】このような状況に鑑みて、マルチ解像度データを有する画像符号化データの中から必要な部分のデータのみを取り出して別の符号化データに変換する方法として従来から様々な方法が提案されており、例えば米国特許第６０４１１４３号（「MULTIRESOLUTION COMPRESSED IMAGE MANAGEMENT SYSTEM AND METHOD」）が挙げられる。この特許では、ソースとなる画像データは、ウェーブレット変換或いはウェーブレットのような変換を用いてマルチ解像度のデータを管理する符号化データである。そして、このソースとなる符号化データの中から、クライアントが選択した空間的な領域のデータを処理するために必要な符号化データを取り出し、１つの独立した符号化データに変換するものである。この時、ソースの符号化データから取り出される部分的な符号化データは、ＪＰＥＧ２０００のコードブロックに対応しており、今回のクライアントからの要求を処理するために必要な符号化データを全ての階層で、かつ全てのサブバンドから該当する、或いは関係する全部の符号化データを含むものである。

【０００６】

【発明が解決しようとする課題】しかし、符号化データをクライアント側で最初から再デコードする場合、サーバ側から送られてくる断片的な符号化データを、直接Ｊ

ＰＥＧ２０００準拠の１つの符号化データファイルに作り直すことも可能ではあるが、サーバ側から送られてくる断片的な符号化データは、クライアント側から要求する順に送られてくるため、この受信した符号化データをＪＰＥＧ２０００準拠の１本の符号化データに変換することは、クライアント側の処理が煩雑になり、コストがかかるという問題点がある。

【０００７】又、サーバから送られてきた断片的な符号化データを独自形式のキャッシュファイルとして構成し、クライアント側のＪＰＥＧ２０００デコーダが直接このキャッシュファイルをリードすることも可能であるが、この場合、ＪＰＥＧ２０００のデコーダは、この独自のキャッシュファイルをリードするための処理が必要となる。このため、処理が複雑になると共に、汎用のＪＰＥＧ２０００デコーダが使えないという問題点がある。

【０００８】そこで、ＪＰＥＧ２０００の画像をパケット単位で受信し、それらパケットデータをキャッシュしておく。そして、デコーダへデータを渡す際に、まだ受信していないパケットデータ部分に、zero length packet（以下、ＺＬＰと略す）データを挿入し、既に受信したパケットデータと合せて、一つのファイルを作成することで、メインヘッダの書き換えなど煩雑な処理を行わずに、一般的なＪＰＥＧ２０００デコーダで処理できるＪＰＥＧ２０００ファイルを作成する方法もある。

【０００９】しかし、受信した全てのパケットデータを利用して作成された符号化データは、クライアント側のアプリケーションを使っているクライアントが要求した画像サイズ、画質および画像領域など、クライアントの望む表示形態を考慮していない。従って、この符号化データを受け取ったデコーダは、その一部のデータから生成されたデコード結果を利用して表示画面を作成することがあり、キャッシュされている全ての符号化データを使って１本のＪＰＥＧ２０００符号化データファイルを作ることは、デコーダ自体の処理にも無駄な処理が生じるという問題がある。

【００１０】更に、キャッシュされている符号化データの量が多くなるほど、デコーダに渡す符号化データを作成するときに発生するデータのコピー作業が多くなる。これは、クライアントの表示に要するまでの時間が長くなることを意味しており、パフォーマンス低下という問題が生じる。特に、画像の拡大やスクロールという命令をクライアントが行うことで、表示に直接必要のないキャッシュデータが多くなり、上記の問題が頻繁に起こる。

【００１１】また、クライアントが表示したい表示方法に最適なProgression Orderは、クライアントの表示要求により変化する。以下、図３０を用いて説明する。

【００１２】図中、３００１は、解像度（Resolution）が４個（Resolution0～3）、レイヤ（Layer）が３個（Layer0～2）ある場合を、Resolution Progression Orderで格納した場合の符号化データを示す。一方、３００２は、符号化データ３００１をSNR Progression Orderで格納した場合のパケットの概念を示す。これら符号化データ及びパケットから、解像度が「０」、レイヤが「０」から「２」までをデコードする場合を考えてみる。符号化データ３００１で示すResolution Progression Orderの場合は、３００３ａで示すように３個のパケットを連続して処理することでデコードできる。一方、パケット３００２で示すSNR Progression Orderの場合は、３００３ｂで示す３個のパケットを処理するためには、３００４と３００５で示す６個のパケットも処理しなければならない。しかし、これまでは、サーバ側が送りだしたＪＰＥＧ２０００ファイルのProgression Orderと、クライアントが表示したい表示方法に最適なProgression Orderとが異なる場合でも、クライアント側で管理しているキャッシュファイルは、サーバ側で管理しているＪＰＥＧ２０００のファイルのProgression Orderと同じであるため、単純にキャッシュファイルから１本のＪＰＥＧ２０００符号化データに変換すると、クライアント側のＪＰＥＧ２０００デコーダに多大な負荷がかかるという問題点がある。

【００１３】更に、上述した米国特許第６０４１１４３には、以下の問題点がある。

【００１４】まず、この特許をネットワークを通した通信であるＩＩＰにそのまま適応した場合、クライアントからの要求の度に、サーバからクライアントに対して全ての階層の全コードブロックの符号化データを送ることになり、サーバはクライアントが既に受信済みの符号化データまでも送信することになる。更に、送信する符号化データのヘッダ部分は、今回要求した画像データの情報（画像サイズ、階層情報など）に書き換えられており、元々サーバで管理している符号化データの情報を取ることができない。

【００１５】本発明は上記従来例に鑑みてなされたもので、断片化された符号化データを受信して格納しておき、その格納している符号化データを用いて表示要求に応じた符号化データを作成することができる符号化データの作成方法及びその装置を提供することを目的とする。

【００１６】又本発明の目的は、高速な符号化処理を可能にした符号化データの作成方法及びその装置を提供することにある。

【００１７】

【課題を解決するための手段】上記目的を達成するために本発明の符号化データの作成方法は以下のような工程を備える。即ち、断片化された符号化データを受信してメモリに格納する格納工程と、ユーザの表示要求を取得する取得工程と、前記取得工程で取得した前記表示要求に基づいてProgression Orderを判断する判断工程と、

前記判断工程での判断に基いてヘッダ情報を変更する変更工程と、前記変更工程で変更された前記ヘッダ情報に基づいて、前記メモリに該当する符号化データが格納されているかどうかを判定する判定工程と、前記判定工程において前記メモリに格納されていると判定された前記断片化された符号化データを読み出して前記表示要求に適したProgression Orderに変換するとともに、前記メモリに格納されていない符号化データにはダミー符号を適用してＪＰＥＧ２０００準拠の符号化データを作成する工程とを有することを特徴とする。

【００１８】上記目的を達成するために本発明の符号化データの作成装置は以下のような構成を備える。即ち、断片化された符号化データを受信して格納する格納手段と、ユーザの表示要求を取得する取得手段と、前記取得手段で取得した前記表示要求に基づいてProgression Orderを判断する判断手段と、前記判断手段での判断に基いてヘッダ情報を変更する変更手段と、前記変更手段で変更された前記ヘッダ情報に基づいて、前記格納手段に該当する符号化データが格納されているかどうかを判定する判定手段と、前記判定手段において前記格納手段に格納されていると判定された前記断片化された符号化データを読み出して前記表示要求に適したProgression Orderに変換するとともに、前記格納手段に格納されていない符号化データにはダミー符号を適用してＪＰＥＧ２０００準拠の符号化データを作成する手段とを有することを特徴とする。

【００１９】

【発明の実施形態】以下、添付図面を参照して本発明の好適な実施の形態を詳細に説明する。

【００２０】尚、本実施の形態は、ネットワークを介して受信した断片的な画像データを、一つの画像ファイルに作成し直す方法に関し、特にＩＳＯ／ＩＥＣ－１５４４４に準拠したＪＰＥＧ２０００符号化データをインターネット・イメージング・プロトコル（ＩＩＰ）などの画像符号化データの断片的な領域、或いは、断片的な符号化データを受信し、それらの断片化された符号化データを独自の方式でキャッシュするクライアントにおいて実施可能であり、更には、クライアントのアプリケーションプログラムにより要求される表示要求に応じて、クライアント側で受信した断片化された符号化データのキャッシュデータから１つのＪＰＥＧ２０００準拠の符号化データファイルを生成するものである。

【００２１】図１は、インターネットに代表されるネットワーク上に複数のコンピュータが接続されている様子を示す概念図である。

【００２２】図１において、１００はインターネットに代表されるネットワークを示し、１０１，１０３はサーバコンピュータであり、それぞれ画像データを送信するためのＪＰＥＧ２０００用のＩＩＰサーバをはじめ、ＷＷＷサーバ機能に必要なソフトウェアを実行しており、またこれらサーバ１０１，１０３はそれぞれ大容量のハードディスク１０１ａ，１０３ａを接続しており、そのハードディスクに大量の画像データも格納している。１０２ａ，１０２ｂはクライアントコンピュータであり、それぞれ、ウェブ（Ｗｅｂ）ブラウザ、ＪＰＥＧ２０００デコーダなど、クライアント側で必要なソフトウェアを実行している。

【００２３】図２は、サーバコンピュータ１０１，１０３或いはクライアントコンピュータ１０２ａ，１０２ｂ等のハードウェア構成を示すブロック図である。

【００２４】図２において、２０１はＣＰＵで、装置全体の制御などを行っている。２０２はキーボードで、マウス２０２ａとともに、この装置に情報などを各種データやコマンド等を入力するために使用される。２０３は表示部で、例えばＣＲＴや液晶ディスプレイなどを備えている。２０４はＲＯＭ、２０５はＲＡＭで、この装置における記憶部を構成し、この装置が実行するプログラムや、この装置が利用するデータ等を記憶している。２０６はハードディスク装置、２０７はフロッピィディスク装置で、これらは、この装置で使用される外部記憶装置を構成している。２０８はプリンタである。２０９はネットワークを制御するネットワーク・インターフェースで、ここからインターネットなどのネットワーク１００に接続されているサーバコンピュータなどのネットワーク上の資源にアクセスしたり（クライアントの場合）、クライアントからの要求に応じて符号化された画像や文書データを送信する（サーバの場合）ことができる。

【００２５】本実施の形態では、既に生成済みのＪＰＥＧ２０００ファイルがサーバコンピュータ１０１，１０３で管理されている。一方、クライアントコンピュータ１０２ａ，１０２ｂでは、画像表示アプリケーションなど、クライアントが要求したＪＰＥＧ２０００ファイル内の必要な部分のみを、ＪＰＥＧ２０００用のＩＩＰを使用してサーバコンピュータ１０１，１０３に要求し、それに応答して送られてくる符号化データを断片的に受信し、この受信した断片的な符号化データをクライアントコンピュータ１０２ａ，１０２ｂの、例えばハードディスク２０６等にキャッシュする。このハードディスク２０６などにキャッシュされた断片的なＪＰＥＧ２０００符号化データからＪＰＥＧ２０００のデコーダに渡すための、ＪＰＥＧ２０００準拠の符号化データに変換する部分を説明する。

【００２６】クライアント・コンピュータ１０２ａ，１０２ｂのユーザは、例えばウインドウズ（登録商標）（Ｗｉｎｄｏｗｓ（登録商標））マシンを使用してホームページを開き、そこに書かれているＪＰＥＧ２０００の画像へのリンクをクリックすることにより、ＪＰＥＧ２０００の画像を、クライアント１０２ａ，１０２ｂの用途に適した画像サイズや解像度で表示するために必要

な断片的なデータとして取得しキャッシュする。そして、これらキャッシュデータから一本のビットストリームを作り出してデコードし、表示するという場合を想定する。

【００２７】次に図３を参照して一般的なＪＰＥＧ２０００の符号化データを説明する。

【００２８】図３は、Layer-resolution level-component-position progression（以下、SNR Progressionと記す）に沿って記録されたＪＰＥＧ２０００ファイルの構成を示す図である。

【００２９】このSNR Progressionに準じた場合、（レイヤ）Layer／（解像度）Resolution／（成分）Component／（位置）Positionの順に記録される。このようなデータの並び方は、Progression Orderと呼ばれる。解像度（画像サイズ）とResolution番号との関係を図４に示す。

【００３０】図４において、最も小さい解像度の画像の解像度（resolution）番号を「０」とし、resolution番号が一つ増加するごとに画像の幅と高さが共に倍になっていく。また各レイヤ（Layer）は、Resolution番号の小さい順に割り当てられている。レイヤ番号は、復元する画像の原画に対するＳ／Ｎ比に対応し、レイヤ番号が小さいほどＳ／Ｎ比が悪くなる。一つのＪＰＥＧ２０００ファイル内でのResolution番号とレイヤ番号、component番号の最大値は、エンコーダによって予め設定され、そのパラメータに従ってエンコードされており、その情報は符号化データの中に格納されている。また各パケット（packet）は、そのパケットに格納されているコードブロック（code-block）の情報を管理しているパケットヘッダ（packet header）部と、各コードブロックの符号化データから構成されている。

【００３１】このようなＪＰＥＧ２０００符号化データを使えば、クライアントはサーバ１０１（以下の説明では例としてサーバ１０１の場合で説明するが、サーバ１０３の場合も同様にして実行されることはもちろんである）にある全ての画像データを取得することなく、必要な部分のデータのみをサーバ１０１から受信することが可能である。クライアントの受信データの単位としては、ＪＰＥＧ２０００のパケット（packet）、或はパケットよりも更に小さい符号化単位であるコードブロック単位が考えられる。本実施の形態では、クライアントがサーバから受信するデータ単位としてパケット単位を想定する。このパケット単位でのリクエスト及びレスポンスの概念図を図５に示す。

【００３２】図５において、クライアント１０２ａはサーバ１０１に画像のタイル番号（Tile）と（resolution level）とレイヤ（Layer）、component、position番号を指定してデータを要求する。図５の例では、タイル番号（Tile）は「１」、解像度「０」、レイヤ「０」、Componentは「０」、そしてPosition番号は「０」である。これによりサーバ１０１は、画像５０３のコードストリームを解析して、その指定されたタイル番号、Resolution levelとLayer、component、position番号に相当するパケット（packet）データを抜き出し、パケット０としてクライアント１０２ａに送り返す。

【００３３】図６は、本実施の形態で使用するサーバ１０１で管理されているＪＰＥＧ２０００ファイルの一例を示す図である。

【００３４】図において、６００は、SNR Progressionでエンコードされている例を示し、Layerは「０」～「２」の３種類、Resolution levelは「０」～「３」の４種類、componentとpositionは各１種類である。この場合の各パケットの順を６０１に示す。また説明を簡単にするために、サーバ１０１で管理しているＪＰＥＧ２０００の最大解像度の画像サイズを１０２４×１０２４画素とする。よって４種類のResolution levelがあると、各解像度での画像サイズは、１０２４×１０２４，５１２×５１２，２５６×２５６，１２８×１２８画素となる。

【００３５】図７は、クライアント１０２ａ（以下、代表して示す）のアプリケーションでの処理の概略を示すフローチャートである。ここでは、クライアント１０２ａでＪＰＥＧ２０００アプリケーションが起動しており、表示したいＪＰＥＧ２０００ファイルが別の手段で指定された後での、画像の表示処理を示すフローチャートである。

【００３６】まずステップＳ１で、ユーザが表示のための操作を行う。アプリケーションの起動時は、例えば開いたウィンドウサイズに収まる画像サイズを計算してもよい。次にステップＳ２に進み、ステップＳ１の操作により新たに必要となったパケットを全て算出する。その後ステップＳ３に進み、ステップＳ２の算出結果により新たに必要となったパケットデータをサーバ１０１に対して要求する。その後ステップＳ４に進み、その要求に応答してサーバ１０１から送られてくる、要求した全てのパケットデータを受信し、それをクライアントコンピュータ１０２ａにキャッシュする。次にステップＳ５に進み、キャッシュ形式で管理している符号化データを、１本のＪＰＥＧ２０００符号化データ形式に変換する。その後ステップＳ６に進み、ステップＳ５で作成したＪＰＥＧ２０００符号化データをデコードして表示する。そしてステップＳ７に進み、ユーザが終了を要求した場合はアプリケーションを終了し、他の表示要求を行った場合は、再度ステップＳ１に戻って前述の処理を実行する。

【００３７】［実施の形態１］次に、具体的な処理内容について説明する。

【００３８】まずクライアント１０２ａでのアプリケーションが起動した直後では、このアプリケーションの画面表示サイズは１２８×１２８画素であるとし、このサ

イズに入るサイズの画像データを表示する場合を考える。更にこの場合、図７のステップＳ２で算出される必要なパケットは、最低解像度の画像サイズで、SNRが最大のデータを表示することを考える。この場合は、図６における６０２の「Layer0／Reso0／Comp0／Pos0」、６０３の「Layer1／Reso0／Comp0／Pos0」、６０４の「Layer2／Reso0／Comp0／Pos0」で示す３個のパケットが必要となる。従ってサーバ１０１のプログラムは、これら３個のパケットを切り出す処理を行う。図６から分かるように、SNR Progressionで格納されていると、６０２と６０３の間、６０３と６０４の間には、今回のデコード処理では必要ないパケットが各３個ずつ入っている。

【００３９】次に図７のステップＳ５の符号化データ生成の詳細な処理を図８乃至図１１のフローチャートを参照して以下に詳しく説明する。

【００４０】図８は、図７のステップＳ５における「符号化データの作成」処理を示すフローチャートである。

【００４１】まずステップＳ１１で、クライアントコンピュータ１０２ａのユーザがアプリケーションに対してどのような表示要求を出したかを取得する。次にステップＳ１２に進み、ステップＳ１１で取得した表示要求から、ＪＰＥＧ２０００のデコーダが高速に処理できるProgression Orderを判断する。この判断処理は、図９のフローチャートを参照して後述する。次にステップＳ１３に進み、ステップＳ１２で判断したProgression Orderと、キャッシュされているProgression Orderとが同一か否かを判断する。同一の場合はステップＳ１４に進み、不一致の場合はステップＳ１５にそれぞれ移動する。ステップＳ１４ではProgression Orderが同一であるため、キャッシュ内の形式通りにコピーしてＪＰＥＧ２０００符号化データファイルを生成する。一方、ステップＳ１５では、Progression Orderを変換しながら、ＪＰＥＧ２０００符号化データファイルを生成する。

【００４２】次に図９を参照して、図８のステップＳ１２における「Progression Orderの判断」処理について説明する。

【００４３】まずステップＳ２１、ステップＳ２２にて、今回、サーバ１０１に要求したＪＰＥＧ２０００符号化データファイルの解像度レベル（resolution level）数、レイヤ（Layer）数を各変数Ro，Loに取得する。次にステップＳ２３，ステップＳ２４では、ステップＳ１１で取得した、ユーザが今回指定した表示要求に基づいて、表示用の解像度レベル数、レイヤ数を、それぞれ変数Rq，Lqに取得する。尚、これら変数Ro，Lo，Rq，Lqは、上述したＲＡＭ２０５に設定されている。その後ステップＳ２５に進み、各変数の値に基づいてRq×Lo，Lq×Roをそれぞれ計算し、その計算結果を比較する。ここでRq×Lo＜Lq×Roの場合はステップＳ２６に進み、それ以外の場合はステップＳ２７に進み、夫々「Resolution Progression」、「SNR Progression」と判断する。

【００４４】具体例を使って説明する。ステップＳ２１とステップＳ２２で、それぞれRo＝４（解像度０～３），Lo＝３（レイヤ０～２）、ステップＳ２３とステップＳ２４で各々Rq＝１（１２８×１２８であるため），Lq＝３（レイヤ０～２）となる。この結果、Rq×Lo＝３，Lq×Ro＝１２となってRq×Lo＜Lq×Roを満足する。これによりステップＳ２６に進み、「Resolution Progression」が選択される。これにより図８のステップＳ１３の判断において、異なるProgression Orderであると判断されてステップＳ１５の変換コピー処理に移動する。

【００４５】図１０は図８のステップＳ１４の「そのままコピー」の詳細処理を示すフローチャート、図１１は図８のステップＳ１５の「変換コピー」の処理の詳細を示すフローチャートである。

【００４６】まず図１０を参照して、「そのままコピー」の処理について説明する。

【００４７】まずステップＳ３１，Ｓ３２で、タイル数とパケット数を取得し、それぞれ変数ｉTMax，ｉPMaxに記憶する。次にステップＳ３３に進み、ＪＰＥＧ２０００のメインヘッダを出力する。次にステップＳ３４に進み、内部変数ｉTを「０」で初期化する。尚、これら変数は、前述したようにＲＡＭ２０５に記憶される。

【００４８】次にステップＳ３５に進み、タイルヘッダ（Tile Header：ＳＯＴマーカコード）を生成し、仮出力する。次にステップＳ３６，Ｓ３７で、パケット数をカウントする変数ｉP，パケットデータのバイト数を計数する変数ｉLを共に「０」に初期化する。次にステップＳ３８に進み、変数ｉPで示されるパケットのデータがキャッシュにあるか否かを判断する。キャッシュにある、即ち、サーバ１０１から転送済みであればステップＳ３９に進み、キャッシュに存在しなければステップＳ４２にそれぞれ移行する。ステップＳ３９では、変数ｉPで示されるパケットデータを出力し、次にステップＳ４０で、その出力したパケットデータのバイト数を変数ｉLに足し込む。一方、キャッシュにない場合はステップＳ４２に進んでＺＬＰ（zero length packet）のデータを出力し、次にステップＳ４３に進み、ＺＬＰのバイト数、即ち、「１」を変数ｉLに足し込む。

【００４９】こうしてステップＳ４０或はステップＳ４３を実行した後ステップＳ４１に進み、変数ｉPをインクリメントし（＋１）、次にステップＳ４４に進んで、変数ｉPと変数ｉPMaxとを比較する。このステップＳ４４は、タイル内の全てのパケット数分の処理を行ったかを判断するものである。ここで全てのパケットを処理していなければ再度ステップＳ３７の処理に戻って前述の処理を実行するが、全てのパケットの処理が終わっていればステップＳ４５に進み、タイル全体のデータ数を管理しているフィールドの値を変数ｉLから更新する。

そして、出力したＳＯＴマーカコード内のタイル全体のデータ数を出力する。次にステップＳ４６に進み、タイル数を計数する変数ｉＴをインクリメント（＋１）する。そしてステップＳ４７に進み、変数ｉＴの値が最大タイル数に到達したか、即ち、全てのタイルの処理が終わったか否かを判断する。終わっていなければステップＳ３５に戻って前述の処理を実行するが、全てのタイルの処理が終わっていればステップＳ４８に進み、ＪＰＥＧ２０００の符号化データを完成させるための処理を行う。ここでは、ＥＯＣマーカコードなどを出力する。

【００５０】図１１は、Progression Orderを変更しながらＪＰＥＧ２０００符号化データを生成する図８のステップＳ１５の「変換コピー処理」を示すフローチャートで、この図１１では、図１０で示す処理と同じ処理を行う部分には同一の番号を付けている。よって、図１０で示した処理と異なる部分のみを説明する。

【００５１】ステップＳ３１，Ｓ３２で、全てのタイル数とパケット数とをそれぞれ変数ｉＴMax，ｉＰMaxに格納した後、ステップＳ５０では、図８のステップＳ１２で判断したＪＰＥＧ２０００のデコーダが高速に処理できるProgression Orderを示すメインヘッダのデータを生成して出力する。具体的には、図６の６００で示すＣＯＤマーカコード内の「Progression Order」のフィールド値を変更する。図１１において、点線５００で囲んである部分の処理が図１０で示したフローチャートと異なっている。即ち、ステップＳ５１では、変数ｉＰで示される順に出力すべきパケットを、ステップＳ５０で変更した「Progression Order」から算出する。その後ステップＳ５２では、実際にそのパケットデータを取得して出力する。残りの部分の処理は図１０で説明したものと同一であるため、その説明を割愛する。

【００５２】次に、本実施の形態に係る処理の具体的な例について説明する。

【００５３】本実施の形態１の場合は、「Resolution Progression」が出力符号化データのProgression Orderとして選択されており、一方、キャッシュ内のProgression Orderは、「SNR Progression」であるために、図１１で示す「変換コピー」処理が選択される。

【００５４】ステップＳ３１，Ｓ３２では、タイル数は「１」であるため変数ｉＴMax＝１とし，図６に示すように１２個のパケットがあるため、変数ｉＰMax＝１２をセットする。次にステップＳ５０では、図１２の１１０５で示すように、ＣＯＤマーカコード内のProgression Orderフィールドを「SNR Progression」を示す値「１」に変更してメインヘッダを書き込む。次にステップＳ３４に進み、変数ｉＴを「０」に初期化し、ステップＳ３５で、タイルヘッダ「Tile Header」を仮に出力する。その後ステップＳ３６，Ｓ３７で、各変数ｉＰ，ｉＴileＬをともに「０」で初期化する。次にステップＳ５１で、変数ｉＰが指す順番に出力するパケットをキャッシュから探す。最初は変数ｉＰの値は「０」なので、１番目に出力するパケット、即ち、「Layer0／Reso0／Comp0／Pos0」のパケットデータを探す。次にステップＳ３８に進み、このパケットデータがキャッシュ内にあるかを判断する。今回は既にキャッシュにあるためステップＳ５２に進み、実際に「Layer0／Reso0／Comp0／Pos0」のパケットデータを取得して出力する。その後ステップＳ４０に進み、その出力したパケットのバイト数を変数ｉＴileＬに足し込む。その後ステップＳ４１に進み、パケット数をカウントする変数ｉＰをインクリメント（＋１）し、続くステップＳ４４で、変数ｉＰと変数ｉＰMaxとを比較する。今回はｉＰ＜ｉＰMaxでないし、即ち、全てのパケットを読込んでいないためステップＳ３７に戻る。そして変数ｉＰの値が「１」，「２」の時は上記と同様の処理を行い、変数ｉＰの値が「３」～「１１」までは、ステップＳ３８の判断によりキャッシュに存在しないと判断されるためステップＳ３８からステップＳ４２に進み、ＺＬＰのデータを出力し、ステップＳ４３で、変数ｉＴileＬをインクリメントする。こうしてステップＳ４１の処理後、変数ｉＰの値が「１２」になるとステップＳ４４の判断によりｉＰ＜ｉＰMaxでなくなってステップＳ４５に進み、変数ｉＴileＬからタイルのバイト数を計算し、ＳＯＴマーカ内の該当する位置を変更する。次にステップＳ４６に進み、変数ｉＴの値をインクリメントし、ステップＳ４７の判断により、全てのタイルを処理したかを判断する。本実施の形態では、タイルの数は１つなので、そのままステップＳ４８に進んで終了処理を行なう。

【００５５】この様にして図１２に示すＪＰＥＧ２０００符号化データが完成する。

【００５６】他の場合として、解像度を最大まで上げ、ＳＮＲ最低の画像を表示する場合を考える。具体的には、解像度レベル「３」，レイヤ「１」までを表示する。この場合、ステップＳ３（図７）で算出される必要なパケットは、

「Layer0／Reso0／Comp0／Pos0」，「Layer0／Reso1／Comp0／Pos0」，「Layer0／Reso2／Comp0／Pos0」，「Layer0／Reso3／Comp0／Pos0」，「Layer1／Reso0／Comp0／Pos0」，「Layer1／Reso1／Comp0／Pos0」，「Layer1／Reso2／Comp0／Pos0」，「Layer1／Reso3／Comp0／Pos0」，

の８個のパケットが実際のデコード処理に必要になる。この場合のステップＳ５の符号化データの作成処理（図７）を説明する。

【００５７】この場合は図９のステップＳ２１とステップＳ２２では、それぞれRo＝４，Lo＝３となり，ステップＳ２３，ステップＳ２４で、各々Rq＝４，Lq＝２となる。従って、ステップＳ２５でRq×Lo＝１２，Lq×Ro＝８となるため、Rq×Lo＜Lq×Roが満足されずにステップＳ２７に進み、「SNR Progression」が選

択される。これによりステップＳ１３の判断で、キャッシュと同一のProgression Orderであると判断され、ステップＳ１４の「そのままコピー処理」に移行する。その後、図１０のフローチャートで示す処理を行い、図１３で示すようなＪＰＥＧ２０００の符号化データを生成する。

【００５８】これにより、クライアント１０２ａからいかなる表示要求が入力されても、デコーダが実際に処理するパケットデータが、できるだけ符号化データの先頭から連続する符号化データとするべく変換でき、デコード処理を高速に行うことができる。

【００５９】［実施の形態２］前述の実施の形態１では、メインヘッダ内のProgression Orderを変更する場合で説明したが、本実施の形態２では、表示に必要なタイルのみを変更する場合で説明する。

【００６０】図１４（Ａ）は、１枚の画像データを１６個のタイルに分割した時の図を示す図である。

【００６１】図１４（Ｂ）は、サーバ１０１で管理されている符号化データの例を示す図である。この符号化データは、前述の実施の形態１と同様に、「SNR Progression」で全タイルのデータが格納されている。本実施の形態２では、まず画像全体を表示するために、全タイルの解像度レベル「０」，レイヤ「０」の各パケットデータが送られて表示された後、タイル５（Tile 5），６（Tile 6），９（Tile 9），１０（Tile 10）の４個のタイルの部分だけ解像度を上げて表示する場合で説明する。この時に要求する解像度レベルは「１」とする。またこの条件でエンコードされ、サーバ１０１に格納されているＪＰＥＧ２０００符号化データファイルの例を図１４（Ｂ）の１４００に示す。この図に示すように、全てのタイルの符号化条件は同一であり、Progression Orderを示すものは、１４０１に示す様に、メインヘッダ内のＣＯＤマーカに記載されている。

【００６２】このファイルに対して先ず、画像全体を表示するための符号化データ、即ち、全タイルの解像度レベル「０」，レイヤ「２」を要求した後、図１４（Ａ）のタイル５，６，９，１０のみを解像度を上げて表示することを考える。この場合、タイル５，６，９，１０の解像度レベル「１」，レイヤ「２」のみを表示する場合を図１５以降を参照して説明する。

【００６３】図１５は、本発明の実施の形態２に係る符号化データの作成処理を示すフローチャートである。

【００６４】まずステップＳ６１で、表示要求をクライアント側のアプリケーションプログラムから取得するが、その時に表示するタイルの情報までも取得することが、前述の実施の形態１と異なっている。次にステップＳ６２に進み、前述のステップＳ１２と同様にして、ステップＳ６１で取得した表示要求から、ＪＰＥＧ２０００のデコーダが高速に処理できるProgression Orderを判断する。次にステップＳ６３～Ｓ６５で、前述のステップＳ３２～Ｓ３４と同様にして、タイル数を取得して変数ｉTMaxに記憶し、ＪＰＥＧ２０００のメインヘッダを出力し、内部変数ｉTを「０」で初期化する。次にステップＳ６６に進み、タイルデータを出力し、ステップＳ６７で、タイル数をカウントする変数ｉTを＋１し、ステップＳ６８で全てのタイルに対する処理が終了したかを調べ、終了していなければステップＳ６６に戻り、終了するとステップＳ６９で終了処理を実行して、この処理を終了する。

【００６５】図１６は、図１５のステップＳ６６のタイル出力処理を示すフローチャートである。

【００６６】図において、まずステップＳ７１では、タイルのヘッダの内、ＳＯＴマーカのみを出力する。このときＳＯＴマーカのバイト数はこの時点では決まっていないので仮に出力する。これは前述の実施の形態１と同じである。次にステップＳ７２に進み、パケット数を計数する変数ｉPを「０」にし、ステップＳ７３で、タイルのヘッダ部の処理を行う。この処理は図１７のフローチャートを参照して後述する。

【００６７】次にステップＳ７４に進み、ステップＳ７３で設定された変数Ｓの値に応じて３種類の処理を切り替える。ここではステップＳ７４で、Ｓ＝０であればステップＳ７６に進んで、全てのパケットデータをＺＬＰに置き換える。ステップＳ７４でＳ≠０であればステップＳ７５に進み、Ｓ＝１かどうかをみる。Ｓ＝１、即ち、メインヘッダで出力したProgression Orderと本タイルのProgression Orderとが一致するときはステップＳ８１に進み、それがキャッシュされているかどうかを調べ、キャッシュされていればステップＳ８２に進み、キャッシュされている各パケットデータをキャッシュで管理されている順に出力する。またステップＳ８１でキャッシュされていない場合はステップＳ７６に進み、ＺＬＰを出力する。

【００６８】一方、ステップＳ７５でＳ＝１でない場合、即ち、Progression Orderが不一致の場合はステップＳ８３に進み、キャッシュされているかどうか調べ、キャッシュされている場合はステップＳ８４に進み、該当するパケットデータをキャッシュから取得して出力する。またステップＳ８３でキャッシュされていない場合はステップＳ７６に進み、ＺＬＰを出力する。その他の処理は、前述の実施の形態１と同じである。

【００６９】次に、図１６のステップＳ７３のヘッダ部の処理について、図１７のフローチャートを参照して説明する。

【００７０】まずステップＳ９１で、タイル数を計数する変数ｉTileLに初期値「０」をセットする。その後ステップＳ９２に進み、今回処理しているタイルが、図１５のステップＳ６１で取得した、デコードに必要なタイルか否かを判断する。このステップで今回処理しているタイルがデコードに必要ない場合はステップＳ９３に

進む。

【００７１】一方、必要なタイルの場合はステップＳ９５に進み、Progression Orderが同じであるかどうかを判断する。この判断は、図１５のステップＳ６２で判断したProgression Orderが、図１５のステップＳ６４で処理したメインヘッダのProgression Orderと同一か否かを判断する。同一の場合はステップＳ９６に進み、異なる場合はステップＳ９８に進み、ＣＯＤマーカを出力する。このＣＯＤマーカの中には、このタイルのProgression Orderを記入するフィールドがあり、このフィールドに、図１５のステップＳ６２で判断したProgression Orderを示す値を書き込む。その後ステップＳ９９に進んでＳＯＤマーカを出力する。尚、このフローチャートにおいて、ステップＳ９３，Ｓ９６，Ｓ９９の処理は全て同一の処理である。その後、各ステップＳ９４，Ｓ９７，Ｓ１００において、それぞれの状態に応じて変数Ｓの値を決定する。

【００７２】図１８は、実際に図１４の符号化データ１４００に対して、本実施の形態２で示す処理を行った場合に出力される符号化データの一例を示す図である。

【００７３】図において、１８００は、出力した符号化データ全体を示し、１８０１はメインヘッダ内のＣＯＤマーカコードの状態を示し、１８０１で示す、このメインヘッダ内部のProgression Orderは、サーバ１０１にあるファイルと同一の値である。またタイル５，６，９，１０以外のタイルデータは、１８０２に示した値を持っている。即ち、今回のデコード及び表示には関係ないタイルとして、全てＺＬＰのデータになっている。また、タイル５，６，９，１０のデータは、先ず１８０３で示す様にタイルヘッダ内にＣＯＤマーカコードを出力し、更に、ＣＯＤマーカコード内のProgression Orderを示す部分には、「Resolution Progression」を示す値がセットされている。これは、今回の表示要求からＲｑ×Ｌｏ＝２×３＝６，Ｌｑ×Ｒｏ＝３×４＝１２となり、図９のステップＳ２５にてＲｑ×Ｌｏ<Ｌｑ×Ｒｏと判定されてステップＳ２６に進み、「Resolution Progression」と判断されるからである。１８０４は、タイル５の実際のデータを示している。

【００７４】これにより、Progression Orderの変更が必要なタイルのみ変更処理を行い、更に、高速に変換処理を行うことができる。尚、このデコード処理は、前述の実施の形態１と同様に高速に行える。

【００７５】以上説明したように、本実施の形態２に係るファイル作成方法を用いれば、ＩＩＰを用いて送られてくるＪＰＥＧ２０００符号化データの断片的なデータをキャッシュし、このキャッシュファイルから、通常のＪＰＥＧ２０００デコーダで処理できるＪＰＥＧ２０００符号化データを作る時に、今回のデコード及び表示に最適なProgression OrderでＪＰＥＧ２０００符号化データを再構築でき、デコード処理における余分なデータ処理を省くことができ、デコード処理を高速に行うことができる。

【００７６】また、デコード及び表示する領域をタイル単位で指示する場合は、そのデコード、表示に必要なタイルのみに対して処理を行うことができるので処理を高速化できる。更には、タイル毎に表示方法を変えるような複雑な表示方法に対しても対応することができる。

【００７７】【実施の形態３】次に本発明の実施の形態３について説明する。尚、この実施の形態３においても、そのハードウェア構成は、前述の実施の形態と同様である。また、この実施の形態３においても、ユーザの要求する表示に必要なデータを、このパケット単位で受信し、表示する場合で説明する。

【００７８】【タイル単位で書き込みデータを選択する方法】図１９は、本発明の実施の形態３に係る処理を説明するフローチャートである。

【００７９】まずステップＳ１１１において、クライアント１０２ａのユーザは、画像全体が表示領域に納まる解像度のデータをサーバ１０１に要求する。例えば、オリジナル画像（abc.jp2）がサーバ１０１にあるとする。この画像（abc.jp2）は、最大解像度でサイズ１０２４×１０２４（画素）、８×８の６４タイルに分割されており、コンポーネント（component）数が「３」で、解像度レベル（resolution level）が「２」、即ち、３つの画像サイズ方向の階層を持っており、２つのレイヤ（Layer）に分けられていると仮定する。それぞれのタイルには図２０（Ａ）に示すように、左上のタイルから順にシーケンス番号が付されている。この時、サーバ１０１のＪＰＥＧ２０００ビットストリームの構成は図２１の２１０１に示すような形になる。

【００８０】サーバ１０１のＪＰＥＧ２０００ビットストリームが、解像度（resolution）スケーラビリティに準じるデータの並びになっているとすると、それぞれのタイルの中は、図２１の２１０２に示すようなパケットの並びとなる。更に、クライアント１０２ａにおける表示領域を２５６×２５６（画素）と仮定すると、この場合、クライアント１０２ａはステップＳ１１１において、画像「abc.jp2」のヘッダデータ２１０３と、解像度（resolution）「０」を構成する全タイルのパケット、つまり、各タイルの中から図２１の２１０４で表されるパケットデータを要求する。即ち、２（レイヤ）×３（成分）×６４（タイル）×１（解像度）＝３８４（パケット）を要求する。次にステップＳ１１２に進み、受信したパケットデータをキャッシュする。即ち、先ほど要求したヘッダデータと３８４（パケット）データをキャッシュする。次にステップＳ１１３に進み、キャッシュデータの中から、既に受信されていたヘッダデータ２１０１とパケットデータとを利用して、ステップＳ１１１で要求された表示画像に応じたＪＰＥＧ２０００準拠の符号化データを作成する。次にステップＳ１１

４に進み、ステップＳ１１３で作成された符号化データをデコードして表示する。

【００８１】次にステップＳ１１５に進み、ユーザは画像の興味のある一部分を最大解像度まで拡大する要求を出す。例えば、解像度レベル（resolution level）２の中心部の領域２００２（図２０（Ｂ））を表示するために、不足しているデータを要求する。つまりＪＰＥＧ２０００では、各解像度が差分データからできているので、タイル２７，２８，３５，３６の解像度レベル（resolution level）１と、解像度レベル（resolution level）２を構成するパケットデータを要求する。即ち、タイル２７，２８，３５，３６のデータの中のパケットデータ２１０５，２１０６（図２１）で表される２（レイヤ）×３（成分）×４（タイル）×２（解像度）＝４８（パケット）を要求することとなる。次にステップＳ１１６に進み、不足分のデータを受信してキャッシュする。つまり、ステップＳ１１５で要求した４８パケットのデータを受信してキャッシュする。

【００８２】そしてステップＳ１１７に進み、既にステップＳ１１２，Ｓ１１６で受信してキャッシュしているデータから、ステップＳ１１５で要求された表示画像に応じたＪＰＥＧ２０００準拠の符号化データを作成する。次にステップＳ１１８に進み、ステップＳ１１７で作成されたデータをデコードして表示する。つまり、図２０（Ａ）の解像度レベル（resolution level）２の領域２００１が表示される。

【００８３】これらステップＳ１１３、ステップＳ１１７における符号化データ作成の動作を図２２のフローチャートを参照して説明する。

【００８４】まずステップＳ１２１では、クライアントの要求している表示形態を知るために、表示に必要なタイルの番号を把握する。例えば、ステップＳ１１６の場合には、ステップＳ１１５での表示要求から、表示に必要なタイルは２７，２８，３５，３６であることを取得する。そしてステップＳ１２２では、サーバ１０１にあるオリジナル画像のタイル数Ｔmax、および各タイルを構成するパケット数Ｐmaxを取得する。これは、既にサーバ１０１から受信したＪＰＥＧ２０００のメインヘッダを解析して求めても良いし、サーバ１０１に問い合わせて求めても良い。

【００８５】次にステップＳ１２３に進み、表示用ファイルつまり、デコーダへ渡す符号化データを書き出すファイルに、サーバ１０１から受信したメインヘッダをそのまま書き込む。次にステップＳ１２４では、タイル番号を計数するための変数カウンタＴに「０」を入れて初期化する。次にステップＳ１２５では、タイルＴのタイルヘッダを表示用ファイルに書き出す。ここでタイル長は、このタイルに含まれるパケットデータのバイト数に左右される。そのため、この時点では、正確なタイル長を知ることはできない。そこで、暫定的な値として、全て「０」のzerolength packet（ＺＬＰ）である場合のＳＯＴマーカを書き込む。このタイルのデータ長は、（タイルヘッダの長さ）＋（タイルに含まれるパケット数）×１（バイト）として計算する。例えば、タイルヘッダに含まれるマーカがＳＯＴマーカのみで、１つのタイルに１８パケットが含まれている場合は、ＺＬＰは長さが１バイトであるので、タイルのデータ長は１４＋１×１８＝３２（バイト）となる。

【００８６】図２３（Ａ）（Ｂ）は、タイル番号が「０」、含まれるパケット数が「１８」であるタイルに対して作成されるＳＯＴマーカの構成例を示す図である。

【００８７】ステップＳ１２６では、カウンタＴに示されるタイル番号をもつタイルが、ステップＳ１２１で取得した、表示に必要なタイル番号に含まれているか否か判断する。含まれていなければ、つまり、表示に必要ないタイルであればＳ１３８に進み、含まれていれば、つまり、表示に必要なタイルであればステップＳ１２７に進む。例えば、ステップＳ１２１で表示に必要なタイル番号として、２７，２８，３５，３６の番号を取得していれば、これら４つの値とカウンタＴの値を比較し、同じであればステップＳ１２７へ、違っていればステップＳ１３８へ進む。

【００８８】ステップＳ１３８では、タイルＴは表示に必要のないタイルであると判断されているので、タイルＴに含まれる全てのパケットデータの部分に、ダミーの値として、ＺＬＰであることを示す１バイトの値「０」を書き出す。例えば、１タイルに含まれるパケット数Ｐmaxの値が「１８」であれば、図２４の２４０２で示すように、１バイトの値「０」を１８バイト分、タイルヘッダ（ＳＯＴマーカ）２１０７の後ろに書き込む。

【００８９】ステップＳ１２７では、パケット番号のカウンタＰに「０」を入れて初期化し、更に、タイルＴにおいて表示に必要なパケットデータのデータ長をカウントするための変数Lengthに「０」を代入して初期化する。次にステップＳ１２８に進み、パケットＰが既に受信されキャッシュされているかどうかを判断する。未受信のパケットデータであればステップＳ１２９に進み、そのパケットデータがキャッシュされていればステップＳ１３０に進む。例えば、タイル２７に関しては、ステップＳ１１３においては、解像度レベル（resolution level）０のパケットデータ２１０４（図２１）しか受信していないので、Ｐ＝０～５に関しては、ステップＳ１３０へ、Ｐ＝６～１７に関してはステップＳ１２９に進むが、ステップＳ１１６において、最大解像度、最大ＳＮＲで、全てのコンポーネントのパケットデータを受信してキャッシュしているので、Ｐ＝０～１７の１８パケットの場合は全てステップＳ１３０へ進むことになる。

【００９０】ステップＳ１２９では、未受信パケットデータの処理であるので、パケットＰのデータとしてＺＬ

Ｐの値、即ち、１バイトの値「０」を表示用ファイルに書き込む。またステップＳ１２８でキャッシュされている場合はステップＳ１３０に進み、パケットＰのデータをキャッシュから取り出し、そのまま表示用ファイルに書き込む。そしてステップＳ１３１に進み、ステップＳ１２９或はＳ１３０でファイルに書き込んだバイト数を変数Lengthに足す。例えば、ステップＳ１２９において、ＺＬＰの値を書き込んだ場合には、変数Lengthには「１」が足され、ステップＳ１３０において、ファイルに書き込んだパケットＰのバイト数が１００バイトなら、変数Lengthには「１００」が足される。そしてステップＳ１３２に進み、パケット数Ｐのカウント値を「１」だけ増やす。そしてステップＳ１３３で、値ＰがPmaxと同じ値かどうか判断する。同じ値であればステップＳ１３４に進み、同じ値でなければステップＳ１２８に戻る。

【００９１】例えば、解像度レベル（resolution level）数が「３」，レイヤ（Layer）数が「２」，component数が「３」のタイルであれば、タイルは１８パケットを含んでいるので、パケット数Ｐの値が「１８」であればステップＳ１３４に進む。そうでなければステップＳ１２８に戻る。ステップＳ１３４では、変数Lengthの値からタイルＴのタイル長を計算し直し、ＳＯＴマーカのタイル長を表すＰsot２３０１（図２３（Ａ））の値を上書きする。例えば、タイルヘッダにＳＯＴマーカのみが含まれ、変数Lengthの値が「２８６」であった場合には、このＰsot２３０１には、値（１４＋Length=）「３００」が上書きされる。即ち、ステップＳ１１３で作成される表示用ファイルのタイル２７，２８，３５，３６のデータ部分に関しては、サーバ１０１のファイルと全く同じ１８パケットのデータ２１０２（図２１）が書き込まれることになり、図２４で示されるようなファイルが作成される。

【００９２】例えば、ステップＳ１１３において作成される表示用ファイルであれば、各タイルは表示に必要であり、２１０４で示される６個のパケットのデータしか受信していないので、表示用ファイルにおける各タイルのデータ部分には、サーバ１０１から受信したパケットデータ２１０４の後ろ、図２１の２１０５，２１０６で示す部分に、１２バイトの「０」が書き込まれた（２５０５，２５０６）図２５に示すファイルが作成される。

【００９３】ステップＳ１３５では、タイル番号のカウンタＴを１だけ増やす。そしてステップＳ１３６では、全てのタイルに関して、上記のことが行われたかどうかを判断する。計数値ＴがTmaxと同じであれば、即ち、全てのタイルに関して処理が終了しているためステップＳ１３７へ進み、Tmaxと異なればステップＳ１２５に戻る。例えば、タイル数が「６４」であれば、カウンタＴの値が「６４」であればステップＳ１３７に進み、カウンタＴの値が「６４」でなければステップＳ１２５に進む。ステップＳ１３７では、ＪＰＥＧ２０００のＥＯＣコードである２バイトの値「０ｘＦＦＤ９」を書き込み、ファイルを閉じる。

【００９４】このような符号化データの作成処理を採用することにより、表示に必要なタイルかどうかを判断するだけで、そのタイルのパケットデータのみが書き込まれ、その他のタイルのデータには、ＺＬＰの値が書き込まれたファイルが簡単に作成できる。しかも、必要なタイルの未受信データ部分にもＺＬＰを利用することで、簡単にＪＰＥＧ２０００準拠の符号化データファイルを作成できる。

【００９５】［実施の形態４］前述の実施の形態３では、タイル単位で表示に必要なデータであるかを判断し、必要であれば、そのタイルに関して受信したデータの全てを表示用ファイルに書き込み、そのタイル内の未受信のパケットに対しては、ダミーのデータとしてＺＬＰを書き込んでいた。そして、表示に必要ではないタイルのデータは全てＺＬＰを書き込んでいた。つまり、表示されるタイルに関しては、受信した全てのデータを書き込んでいたので、表示の際に必要とされないパケットデータも含まれる可能性がある。従って、ユーザの表示要求に応じたファイルの作成という点ではまだ不十分である。

【００９６】例えば、地図などのように、川崎市全体の地図を表示し、その後、「中原区今井上町５３」というような住所で、その目的地周辺の詳しい場所を、最も解像度の高い画像で把握した後、画面を引いて最寄駅である「武蔵小杉駅」と目的地との位置関係を把握するような場合、前述の実施の形態３に係る方法では、表示に必要の無いパケットデータも含まれてしまう。従って、より表示要求と密接になった符号化データファイルを作成するためには、タイル単位でのデータの取捨選択に加えて、表示に必要なタイル内においてパケット単位でのデータの取捨選択を行う必要がある。

【００９７】［パケット単位で書き込みデータを判断する方法］このように、一度、画像の一部分を最大解像度まで拡大した後、解像度を下げて表示範囲を広げるような動作は図２６に示すステップＳ１４９～Ｓ１５２で表される。尚、この図２６のフローチャートにおいて、ステップＳ１４１～Ｓ１４８の処理は、図１９のフローチャートにおけるステップＳ１１１～Ｓ１１８の処理と同様であるため、その説明を省略する。

【００９８】ステップＳ１４９では、ステップＳ１４８で表示した解像度よりも低い解像度で画像を表示し、表示領域に納まる画像の部分領域を広くするような要求を出す。例えば、ステップＳ１４８において、２５６×２５６（画素）の表示領域に、図２０（Ａ）に示す領域２００１を、解像度レベル２（resolution level 2）で表示していたとすると、ステップＳ１４９では、タイル１８，２１，４２，４５で囲まれる領域を解像度レベル１

（resolution level 1）で表示する、つまり領域２００２（図２０（Ｂ））を表示するように、ステップＳ１４９において要求を出す。次にステップＳ１５０に進み、ステップＳ１４９で要求された表示に必要なパケットで、キャッシュされていないデータを受信してキャッシュする。つまり、タイル番号１８～２１，２６，２９，３４，３７，４２～４５の１２個のタイルに含まれる解像度レベル（resolution level）１を構成するパケットデータ２１０５を受信してキャッシュする。従って、ステップＳ１５０では、２（レイヤ）×３（成分）×１２（タイル）×１（解像度）＝７２（パケット）を受信してキャッシュする。次にステップＳ１５１に進み、キャッシュデータから、表示に必要なパケットのみを使って符号化データを作成する。そしてステップＳ１５２に進み、ステップＳ１５１で作成された符号化データをデコードして表示する。尚、ステップＳ１５１では、キャッシュデータから必要なパケットデータのみから符号化データを作成する。

【００９９】図２７は、図２６のステップＳ１５１の符号化データ作成時の処理動作を示すフローチャートである。尚、この図２７におけるステップＳ１６１～Ｓ１６７，ステップＳ１６９乃至Ｓ１７８の処理は、図２２のステップＳ１２１乃至Ｓ１２７，ステップＳ１２９乃至Ｓ１３８の処理と同じであるため、その説明を省略する。

【０１００】まずステップＳ１６１では、クライアントの表示要求として、タイル番号の他に解像度レベル（resolution level）、レイヤ（Layer）、コンポーネント（component）を取得する。前述の前述の実施の形態３では、タイル番号のみでファイルにデータを書き込むか否かを判断していたが、本実施の形態４では、表示に必要なタイルの中から、更に表示に必要なパケットデータのみを表示用ファイルに書き込む。そのため、解像度レベル、レイヤ、成分といった表示の細かい条件が必要となる。そしてステップＳ１６２以降の処理で、前述の実施の形態３と異なるのは、ステップＳ１６８の処理であり、その他のステップの動作は、実施の形態３と同じであるので、その説明を省略する。

【０１０１】ステップＳ１６８では、表示に必要なタイルに含まれるパケットに関して、パケットＰが要求された表示画面を作成するのに必要かどうかを判断する。必要なパケットデータであればステップＳ１７０に進み、不必要なパケットデータであればステップＳ１６９に進む。例えば、タイル１８について、図２６のステップＳ１５０において、解像度レベル０，１を構成するパケットデータしかキャッシュされていないので、タイル１８の場合には、キャッシュされているパケットデータは、全てステップＳ１６８を経てステップＳ１７０に進んで表示用ファイルに書込まれる。しかし、タイル２７に関しては、キャッシュされた全てのデータを表示画面の作成時に利用するわけではない。解像度レベル０又は１を作成するパケットデータＰ＝０～１１に関しては、ステップＳ１６８において必要であると判断されてステップＳ１７０に進むが、解像度レベル２に相当するパケットデータ（Ｐ＝１２～１７）は、キャッシュデータに存在するが、今回の表示には利用されないので、ステップＳ１６９に進み、そのキャッシュされているパケットデータの代わりにＺＬＰの値、即ち、１バイトの値「０」がファイルに書き込まれる。従って、タイル２７，２８，３５，３６に関しては、図２８に示すようなデータ２８０１が表示用ファイルに書き込まれる。

【０１０２】図２８は、キャッシュされているタイル２７のパケットデータから表示用のデータ２８０１を作成するための処理を説明する概念図である。

【０１０３】ここでは、解像度０，１で、レイヤが１，２、及びコンポーネント０～２であるタイル２７のパケットデータがキャッシュされており、これがそのまま表示用ファイル２８０１として書込まれている。また、解像度２以上のデータは表示には使用されないので、ＺＬＰの値がその後に書込まれている。

【０１０４】図２９は、表示用ファイルにパケットデータが書き込まれる様子を説明する図である。

【０１０５】図２９において、タイル１８に関しては、解像度レベル０，１を構成するパケットデータしかキャッシュされていないので、このタイル１８の場合には、キャッシュされているパケットデータは、全て表示用ファイルに書込まれる。

【０１０６】一方、タイル２７に関しては、キャッシュされた全てのデータを表示画面の作成時に利用するわけではないので、解像度レベル０又は１を作成するパケットデータＰ＝０～１１に関しては必要であると判断されて表示用ファイルに書込まれるが、解像度レベル２に相当するパケットデータＰ＝１２～１７は表示には利用されないので、キャッシュされているパケットデータの代わりにＺＬＰの値、即ち、１バイトの値「０」が書き込まれることになる。

【０１０７】このような符号化データの作成方法を採ることにより、表示に必要なタイルかどうかを判断するだけでなく、更に、そのタイルの中の必要なパケットデータのみが書き込まれ、その他のパケットデータにはＺＬＰの値が書き込まれたＪＰＥＧ２０００準拠の符号化データファイルが簡単に作成できる。

【０１０８】以上の説明において、データのキャッシュ方法については、それぞれのパケットデータが必要なときに取り出せる形式であれば、それぞれのパケットを独立に管理しても、一つのファイルの中にまとめて保存されていても、その他どのような方法でも構わない。

【０１０９】以上説明したように、本実施の形態に係るファイル作成方法を用いることにより、ＩＩＰを用いて送られてくるＪＰＥＧ２０００符号化データの断片的な

データをキャッシュし、このキャッシュしたファイルから、通常のＪＰＥＧ２０００デコーダで処理できるＪＰＥＧ２０００符号化データを作る時に、今回、デコード／表示する部分のみの符号化データをコピーした必要十分な表示用ファイルを作成することが可能である。

【０１１０】これにより、１本の符号化データを作る処理において、データのコピー作業を削減でき、高速なファイル作成が可能になるという利点がある。

【０１１１】また、表示画面を作成するために符号化データをデコードする際に、デコード／表示に必要のないパケットデータ部分にダミーの値としてＺＬＰのデータを利用することで、特別なデコーダを必要とせずに表示処理ができるという利点がある。

【０１１２】更に、表示画面を作成するために、デコーダ側で受け取った符号化データを取捨選択する必要がなくなるという利点がある。また、作成するＪＰＥＧ２０００の符号化データファイルのサイズを小さくすることもできる。更に、ＪＰＥＧ２０００のデコーダも、余計な部分を処理することがなく、高速に処理することもできる。

【０１１３】また本発明は、複数の機器（例えばホストコンピュータ，インタフェイス機器，リーダ，プリンタなど）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機，ファクシミリ装置など）に適用してもよい。

【０１１４】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはＣＰＵやＭＰＵ）が記憶媒体に格納されたプログラムコードを読出し実行することによっても達成される。

【０１１５】この場合、記憶媒体から読出されたプログラムコード自体が前述した実施形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。

【０１１６】プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、ＣＤ－ＲＯＭ，ＣＤ－Ｒ，磁気テープ、不揮発性のメモリカード、ＲＯＭなどを用いることができる。

【０１１７】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているＯＳ（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれる。

【０１１８】さらに、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるＣＰＵなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【０１１９】

【発明の効果】以上説明したように本発明によれば、断片化された符号化データを受信して格納しておき、その格納している符号化データを用いて表示要求に応じた符号化データを作成することができる。

【０１２０】又本発明によれば、高速な符号化処理を可能にできるという効果がある。

【図面の簡単な説明】

【図１】本発明の実施の形態に係るサーバ，クライアントを含むネットワークシステムの構成を示す概念図である。

【図２】本発明の実施の形態１に係るサーバ，クライアントの構成を示すブロック図である。

【図３】本実施の形態に係るＪＰＥＧ２０００符号化データの概念図である。

【図４】ＪＰＥＧ２０００の解像度スケーラビリティを説明する図である。

【図５】本実施の形態に係るサーバ，クライアント間でのＩＩＰのプロトコルの概念図である。

【図６】本実施の形態に係るサーバに格納されている符号化データの一例を示す図である。

【図７】本実施の形態に係るクライアントでの処理を説明するずるためのフローチャートである。

【図８】図７のステップＳ５の符号化データ作成処理を説明するフローチャートである。

【図９】図８のステップＳ１２のProgression Orderの判断処理を説明するフローチャートである。

【図１０】図８のステップＳ１４の「そのままコピー」処理を説明するフローチャートである。

【図１１】図８のステップＳ１５の「変換コピー」処理を説明するフローチャートである。

【図１２】本発明の実施の形態１に係る処理結果の一例を示す図である。

【図１３】本発明の実施の形態１のその他の場合での処理結果の一例を示す図である。

【図１４】本発明の実施の形態２に係るタイル構成の一例を説明する図である。

【図１５】本発明の実施の形態２に係る符号化データ作成処理を示すフローチャートである。

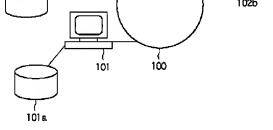
【図１６】図１５のステップＳ６６のタイル出力処理を示すフローチャートである。

【図１７】図１６のステップＳ７３のヘッダ出力処理を示すフローチャートである。

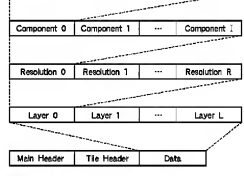
【図１８】本発明の実施の形態２における処理結果の一例を示す図である。

【図１９】本発明の実施の形態３に係るクライアントの画像要求動作を示すフローチャートである。
【図２０】ＪＰＥＧ２０００の解像度スケーラビリティとタイル分割の一例を示す図である。
【図２１】ＪＰＥＧ２０００のパケット単位による符号データの構成例を示す図である。
【図２２】実施の形態３に係る表示用ファイルの符号化データ作成処理を示すフローチャートである。
【図２３】ＳＯＴマーカの構成を説明する図である。
【図２４】作成された表示用ファイルにおける非表示部分のタイルの符号化データの状態を説明する図である。
【図２５】解像度レベル（resolution level）０を表示する際の表示用ファイルにおける表示タイルの符号化データの状態を説明する図である。
【図２６】本実施の形態に係るクライアントの画像要求動作を示すフローチャートである。
【図２７】実施の形態３における表示用ファイルの符号化データ作成処理を示すフローチャートである。
【図２８】実施の形態４におけるキャッシュデータと表示用ファイルデータを説明する図である。
【図２９】本発明の実施の形態４に係る表示用ファイルデータ作成の様子を説明する図である。
【図３０】従来技術を説明するためのＪＰＥＧ２０００符号化データの構成図である。

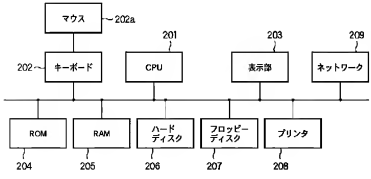
【図１】



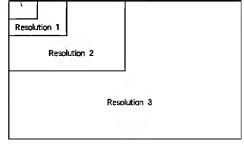
【図３】



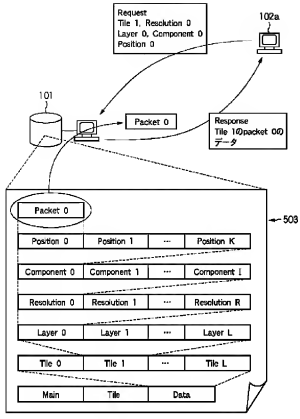
【図２】



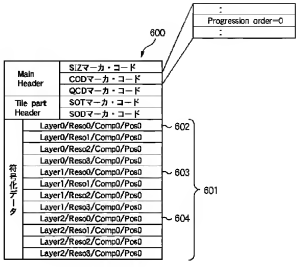
【図４】



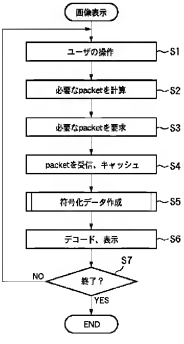
【図５】


Request
Tile 1, Resolution 0
Layer 0, Component 0
Position 0
102a
101
Packet 0
Response
Tile 1のPacket 00)
データ
Packet 0
503
Position 0
Position 1
...
Position K
Component 0
Component 1
...
Component I
Resolution 0
Resolution 1
...
Resolution R
Layer 0
Layer 1
...
Layer L
Tile 0
Tile 1
...
Tile L
Main
Tile
Data

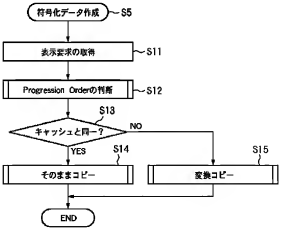
【図６】


600
:
Progression order=0
:
Main Header
SIZマーカ・コード
CODマーカ・コード
QCDマーカ・コード
Tile part Header
SOTマーカ・コード
SODマーカ・コード
タイル内容
Layer0/Reso0/Comp0/Pos0
Layer0/Reso1/Comp0/Pos0
Layer0/Reso2/Comp0/Pos0
Layer0/Reso3/Comp0/Pos0
Layer1/Reso0/Comp0/Pos0
Layer1/Reso1/Comp0/Pos0
Layer1/Reso2/Comp0/Pos0
Layer1/Reso3/Comp0/Pos0
Layer2/Reso0/Comp0/Pos0
Layer2/Reso1/Comp0/Pos0
Layer2/Reso2/Comp0/Pos0
Layer2/Reso3/Comp0/Pos0
602
603
604
601

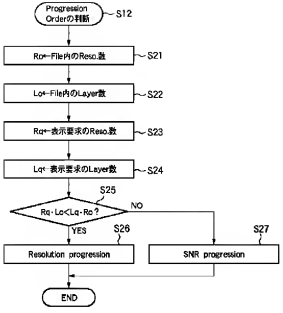
【図７】


画像表示
ユーザの操作
S1
必要なpacketを計算
S2
必要なpacketを要求
S3
packetを受信、キャッシュ
S4
符号化データ作成
S5
デコード、表示
S6
S7
終了？
NO
YES
END

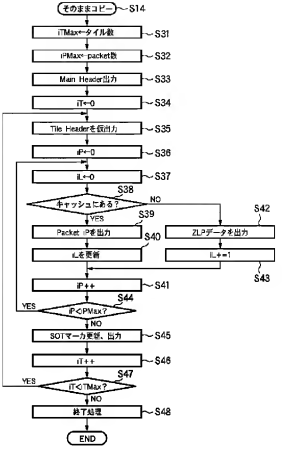
【図８】


符号化データ作成
S5
表示要求の取得
S11
Progression Orderの判断
S12
S13
キャッシュと同一？
NO
YES
S14
そのままコピー
S15
変換コピー
END

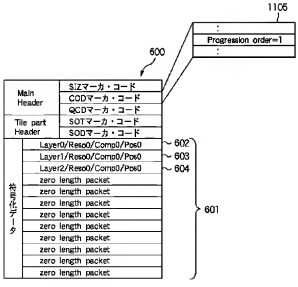
【図９】



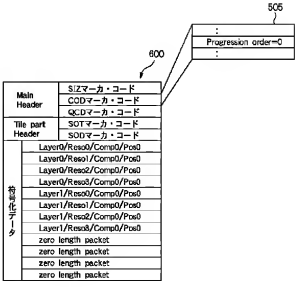
【図１０】



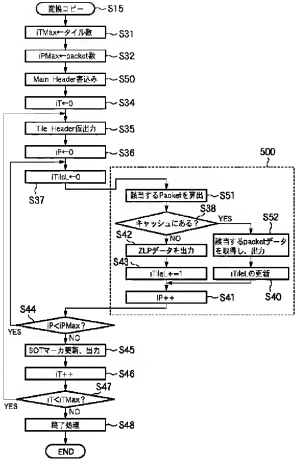
【図１２】



【図１３】



【図１１】

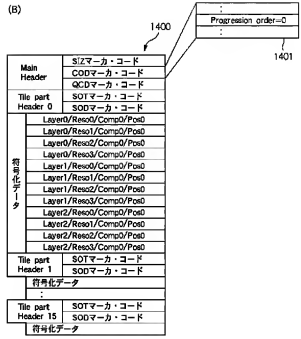


【図１４】

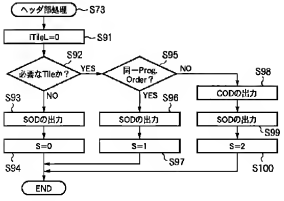
(A)

Tile 0	Tile 1	Tile 2	Tile 3
Tile 4	Tile 5	Tile 6	Tile 7
Tile 8	Tile 9	Tile 10	Tile 11
Tile 12	Tile 13	Tile 14	Tile 15

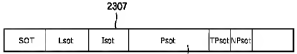
(B)



【図１７】



【図２３】



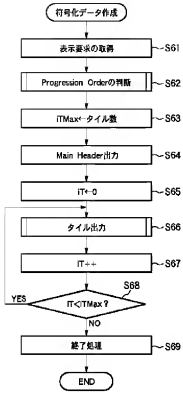
(A)

Parameter	Size(bits)	Values
SOT	16	0xFF90
Lsot	16	10
Isot	16	0
Psot	32	32
TPsot	8	0
NPsot	8	1
	16	0xFF94

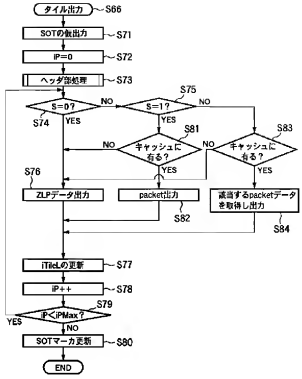
2301

(B)

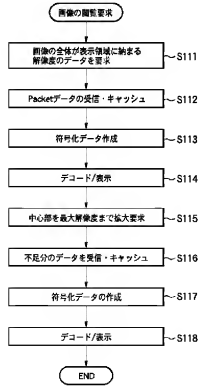
【図１５】



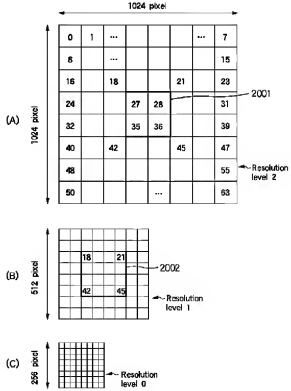
【図１６】



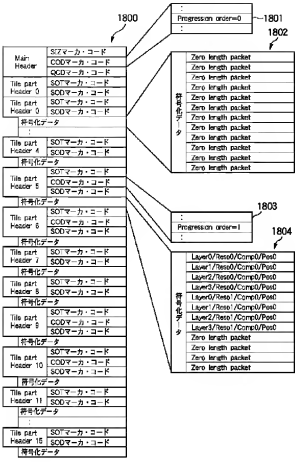
【図１９】



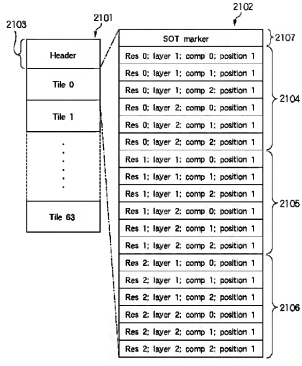
【図２０】



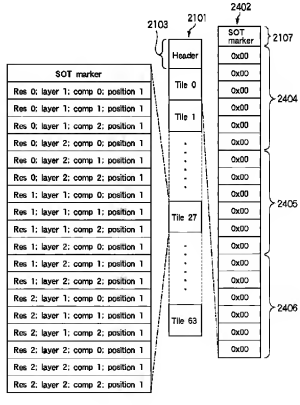
【図１８】


1800
1801
1802
1803
1804
Progression order=0
Progression order=1
Main Header
SIZマーカ・コード
CODマーカ・コード
QCDマーカ・コード
Tile part Header 0
SOTマーカ・コード
SODマーカ・コード
Tile part Header 4
Tile part Header 5
Tile part Header 6
Tile part Header 7
Tile part Header 8
Tile part Header 9
Tile part Header 10
Tile part Header 11
Tile part Header 15
符号化データ
Zero length packet
Layer0/Res0/Comp0/Pos0
Layer1/Res0/Comp0/Pos0
Layer2/Res0/Comp0/Pos0
Layer0/Res0/Comp0/Pos0
Layer0/Res0/Comp0/Pos0
Layer1/Res0/Comp0/Pos0
Layer2/Res0/Comp0/Pos0
Layer3/Res0/Comp0/Pos0
ヘッダ部
符号化データ

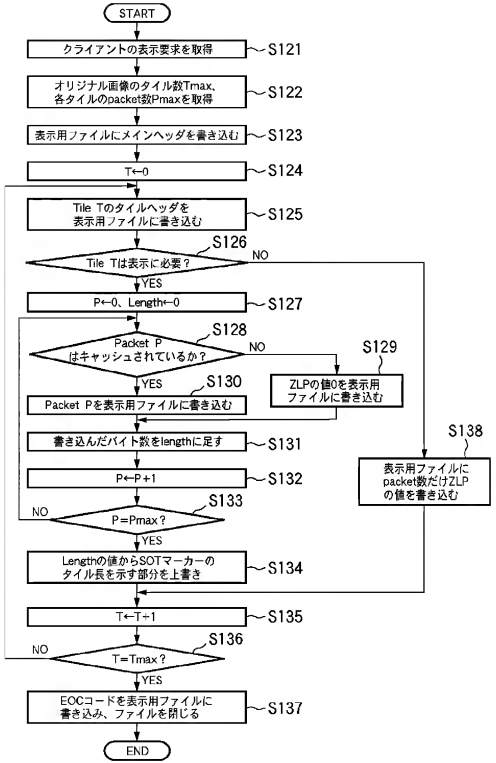
【図２１】


2102
2103
2101
Header
Tile 0
Tile 1
Tile 63
SOT marker
Res 0; layer 1; comp 0; position 1
Res 0; layer 1; comp 1; position 1
Res 0; layer 1; comp 2; position 1
Res 0; layer 2; comp 0; position 1
Res 0; layer 2; comp 1; position 1
Res 0; layer 2; comp 2; position 1
Res 1; layer 1; comp 0; position 1
Res 1; layer 1; comp 1; position 1
Res 1; layer 1; comp 2; position 1
Res 1; layer 2; comp 0; position 1
Res 1; layer 2; comp 1; position 1
Res 1; layer 2; comp 2; position 1
Res 2; layer 1; comp 0; position 1
Res 2; layer 1; comp 1; position 1
Res 2; layer 1; comp 2; position 1
Res 2; layer 2; comp 0; position 1
Res 2; layer 2; comp 1; position 1
Res 2; layer 2; comp 2; position 1
2107
2104
2105
2106

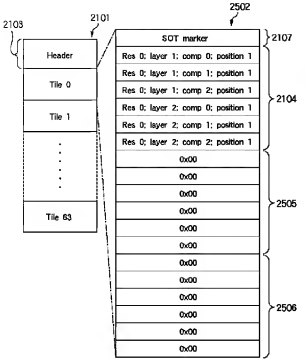
【図２４】


2402
2103
2101
SOT marker
Header
Tile 0
Tile 1
Tile 27
Tile 63
0x00
2107
2404
2405
2406

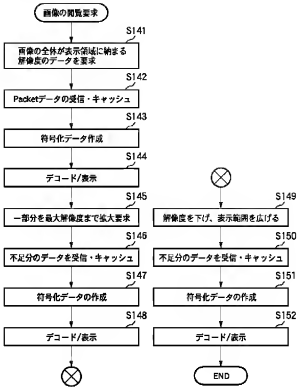
【図２２】


START
クライアントの表示要求を取得
S121
オリジナル画像のタイル数Tmax、
各タイルのpacket数Pmaxを取得
S122
表示用ファイルにメインヘッダを書き込む
S123
T←0
S124
Tile Tのタイルヘッダを
表示用ファイルに書き込む
S125
S126
Tile Tは表示に必要？
NO
YES
P←0、Length←0
S127
S128
Packet P
はキャッシュされているか？
NO
S129
YES
S130
ZLPの値0を表示用
ファイルに書き込む
Packet Pを表示用ファイルに書き込む
書き込んだバイト数をlengthに足す
S131
S138
表示用ファイルに
packet数だけZLP
の値を書き込む
P←P+1
S132
S133
NO
P＝Pmax？
YES
Lengthの値からSOTマーカーの
タイル長を示す部分を上書き
S134
T←T+1
S135
S136
NO
T＝Tmax？
YES
EOCコードを表示用ファイルに
書き込み、ファイルを閉じる
S137
END

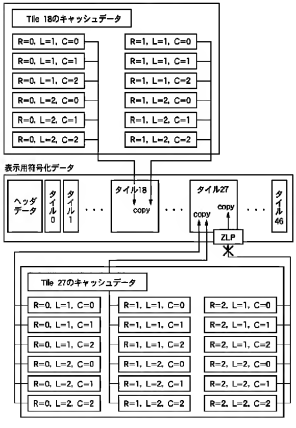
【図２５】



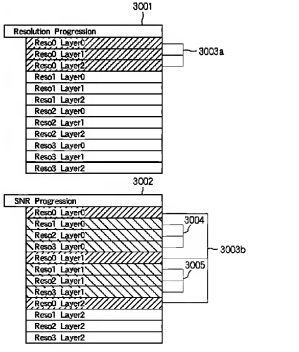
【図２６】



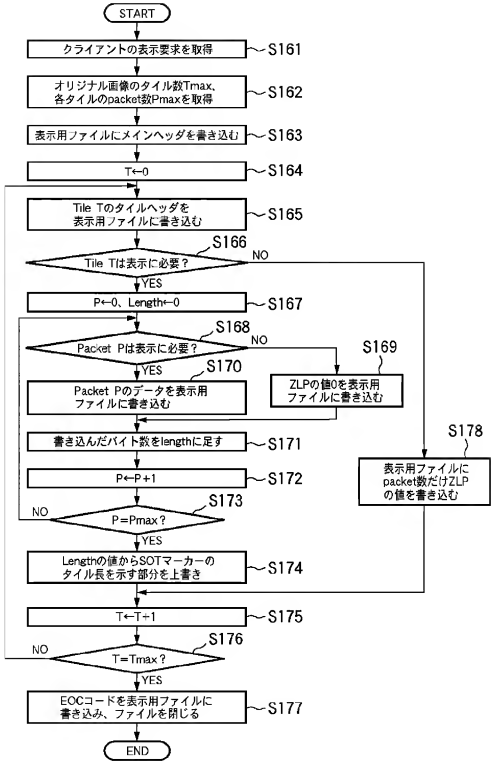
【図２９】



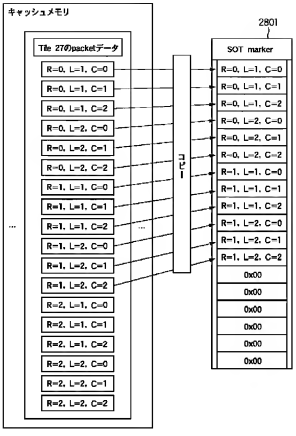
【図３０】



【図２７】


START
クライアントの表示要求を取得
S161
オリジナル画像のタイル数Tmax、
各タイルのpacket数Pmaxを取得
S162
表示用ファイルにメインヘッダを書き込む
S163
T←0
S164
Tile Tのタイルヘッダを
表示用ファイルに書き込む
S165
S166
Tile Tは表示に必要？
NO
YES
P←0、Length←0
S167
S168
Packet Pは表示に必要？
NO
S169
YES
S170
ZLPの値0を表示用
ファイルに書き込む
Packet Pのデータを表示用
ファイルに書き込む
S178
書き込んだバイト数をlengthに足す
S171
表示用ファイルに
packet数だけZLP
の値を書き込む
P←P+1
S172
S173
NO
P＝Pmax？
YES
Lengthの値からSOTマーカーの
タイル長を示す部分を上書き
S174
T←T+1
S175
S176
NO
T＝Tmax？
YES
EOCコードを表示用ファイルに
書き込み、ファイルを閉じる
S177
END

【図２８】



フロントページの続き

Ｆターム(参考)　5C059 KK11 MA00 MA24 MA32 MA34
PP01 RA09 RB02 RE07 SS08
SS20 TA00 TB00 TC01 TC27
TC45 UA02
5C078 BA57 CA34 CA39 EA01
5J064 AA03 BA16 BB04 BC01 BC27

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-169216
(43)Date of publication of application : 13.06.2003

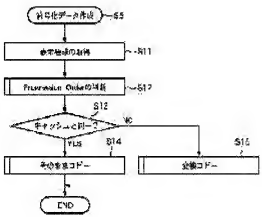
(51)Int.Cl. **H04N 1/41**
H03M 7/30
H04N 7/30

(21)Application number : 2001-364894
(22)Date of filing : 29.11.2001

(71)Applicant : CANON INC
(72)Inventor : ENOKIDA MIYUKI
ISHIKAWA CHIE

(54) METHOD AND APPARATUS FOR CREATING CODED DATA

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a method and apparatus for creating coded data that receives segmented coded-data, caches them and uses the cached coded-data to create coded-data meeting a display requirement.
SOLUTION: The method and apparatus for creating coded data of this invention includes steps of receiving the segmented coded-data to store them to a memory and judging the Progression Order on the basis of a display requirement of a user (S12), revising header information on the basis of the judgment, deciding whether or not the memory stores the corresponding coded data on the basis of the revised header information, reading the segmented coded-data decided to be stored in the memory, converting the read data into the Progression Order suitable for the display requirement and applying a ZLP (zero length packet) process to coded data not stored in the memory to create the coded data in compliance with the JPEG 2000.



LEGAL STATUS

[Date of request for examination] 11.06.2004

[Date of sending the examiner's decision of rejection]	
[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]	
[Date of final disposal for application]	
[Patent number]	3768866
[Date of registration]	10.02.2006
[Number of appeal against examiner's decision of rejection]	
[Date of requesting appeal against examiner's decision of rejection]	
[Date of extinction of right]	